

# Détection de structures communautaires dans des réseaux

SENE, CÔME, PRALON

20 Octobre 2022



## 1 Modularité

## 2 Méthode de Louvain

## 3 Expériences numériques

## 4 Bibliographie



# Définitions et hypothèses

## Représentation

Soit  $G = (V, E)$ , un graphe simple tel que

$V = \{v_1, \dots, v_p\}$  l'ensemble des nœuds

$E \subset \{(v_i, v_j)_{i,j \in \{1, \dots, p\}} | i \neq j\} = V \times V$  l'ensemble des arêtes du graphe

## Densité d'un graphe

On appelle densité d'un graphe la valeur

$$D_G = \frac{|E|}{\frac{p^2 - p}{2}}$$

Correspond à la fréquence d'arêtes dans le graphe, elle rend compte de la connexion entre les nœuds



# Définitions et hypothèses

## Degré d'un noeud

On appelle degré d'un nœud  $i$  la valeur

$$d_i = |\{(v_i, v_j) \in E | j \in \{1, \dots, p\}\}|$$

et correspond au nombre de voisins du noeud  $i$ .

## Modèle nul

On appelle modèle nul d'un graph  $G$ , le graph  $G^*$  dont les  $|E| = m$  arêtes ont été distribuées aléatoirement entre les nœuds de  $G$

Le modèle nul joue un modèle de référence pour lequel il n'existe aucune structure communautaire dans le réseau.



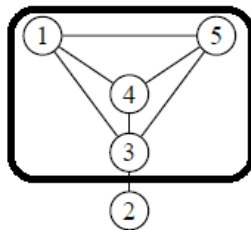


Figure – Détection d'une bonne partition en 2 classes

# Modularité

## Modularité

Soit  $(C_1, \dots, C_K)$  une partition de  $V$ . On définit la modularité  $Q$  de la partition comme

$$Q(C_1, \dots, C_K) = \frac{1}{2m} \sum_{k=1}^K \sum_{(v_i, v_j) \in C_k} (1_{(v_i, v_j) \in E} - P_{i,j})$$

avec  $2m = \sum_{i=1}^p d_i$  et  $P_{i,j} = \frac{d_i d_j}{2m}$

la probabilité que  $v_i$  et  $v_j$  soient connectés sous le modèle nul



1 Modularité

2 Méthode de Louvain

3 Expériences numériques

4 Bibliographie



# Méthode de Louvain

## Etape 1 de l'algorithme

Il s'agit d'un algorithme itératif approchant la partition de modularité maximale en complexité temporelle  $\mathcal{O}(n \log(n))$

- Choix d'un parcours aléatoire
- Affécter chaque noeud à sa propre communauté
- Agrégation de  $i$  à la communauté du voisin  $j$  si

$$\Delta Q = Q(C_1, \dots, C_i, \dots, C_j, \dots, C_p) - Q(C_1, \dots, C_{i,j}, \dots, C_p) > 0$$

et plus grand que les autres voisins de  $i$

- itération sur le chemin





# Méthode de Louvain

## Etape 2 de l'algorithme

- Agrégation des noeuds en communauté
- On obtient un nouveau réseau, pondéré
- Itération de l'étape 1
- Arrêt lorsque  $\Delta Q = 0$



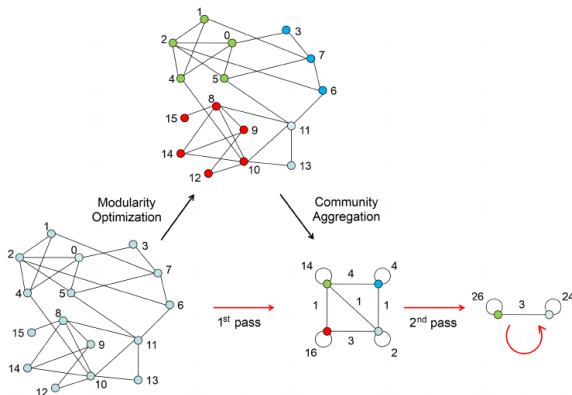


Figure – Exemple d'application de la méthode de Louvain

- 1 Modularité
- 2 Méthode de Louvain
- 3 Expériences numériques
- 4 Bibliographie



# Expériences numériques

## Environnement et librairies utilisées

- Langage : Python 3.9
- Environnement : Spyder
- Librairies : *igraph*, *matplotlib*, *random*

## Graphes utilisés

- Graphe du club de karaté de Zachary
- Graphe généré aléatoirement avec le modèle de *Erdos Rényi*



# La fonction *DetectCommunities*

## Son objectif

Détecter les communautés d'un graphe donné

## Ses arguments

Un objet de type *igraph.Graph* (graphe donné par l'utilisateur)

## Ce qu'elle retourne

Un graphe avec ses communautés mises en évidence par des couleurs



# la fonction *community\_edge\_betweenness* de *igraph*

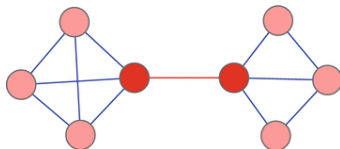
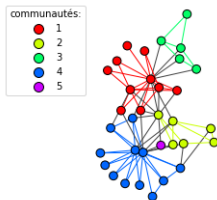
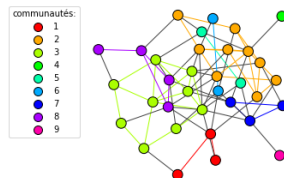


Figure – Exemple d'arête ayant une centralité élevée

# Expériences numériques



(a) communautés détectées pour le graphe "Zachary"



(b) communautés détectées pour le graphe généré aléatoirement

1 Modularité

2 Méthode de Louvain

3 Expériences numériques

4 Bibliographie





- 1 WikiStat. An introduction to network inference and mining, *Article*  
[http://www.nathalievialaneix.eu/doc/pdf/wikistat-network\\_compiled.pdf](http://www.nathalievialaneix.eu/doc/pdf/wikistat-network_compiled.pdf)
- 2 PNAS. Modularity and community structure in networks (2015), *Article*  
<https://www.pnas.org/doi/10.1073/pnas.0601602103#abstract>
- 3 Wikipédia (2022). Méthode de Louvain, *Article*  
[https://fr.wikipedia.org/wiki/Methode\\_de\\_Louvain](https://fr.wikipedia.org/wiki/Methode_de_Louvain)
- 4 igraph, *Documentation*  
<https://igraph.org/python/versions/latest/>
- 5 igraph, *Documentation*  
[https://igraph.org/python/versions/latest/tutorials/visualize\\_communities/visualize\\_communities.html](https://igraph.org/python/versions/latest/tutorials/visualize_communities/visualize_communities.html)
- 6 igraph, *Tutoriel*  
[https://igraph.org/python/api/latest/igraph.\\_igraph.GraphBase.html#Erdos\\_Renyi](https://igraph.org/python/api/latest/igraph._igraph.GraphBase.html#Erdos_Renyi)

